Smooth velocity tracking and kinematic-based fuzzy control of carrying mobile robots

Elham Almodarresi¹

Received: 13 July 2020 / Accepted: 11 November 2020 / Published online: 21 November 2020
© Springer Nature Switzerland AG 2020

Abstract

Mobile robots are affected by forces imposed on them because of moving. These forces significantly affect the carrying efficiency of robots. In this study, the velocity profile is designed such that no extremum emerges on the rising and falling phases of the graph while a constant velocity is preserved between these parts. Decreasing the motion forces on the robot is the main goal of this study. In the second part of this study, a kinematic-based approach is presented to design fuzzy control rules. The significance of this approach is that unlike the methods proposed in the literature, in this method, the rate change of posture error is not required to enter into the fuzzy logic to provide the velocity tracking. This considerably reduces the computational effort with a decrease in the number of rules. In this method, proper indicators are defined as weights of the input and output linguistic functions of the fuzzy logic. Then, the inverse kinematics is employed to compute the proper weights of inputs based on each possible combination of outputs weights.

Keywords Carrying mobile robot · Smooth velocity · Inverse kinematics · Fuzzy control

1 Introduction

After perception and localization, the main step in controlling an autonomous drive robot is trajectory planning. A trajectory planning involves two parts: path planning and velocity planning. A general overview of recent path planning schemes, different categories, and approaches is presented in [1]. After designing the reference trajectory, an appropriate control strategy is required to track the trajectory. Fuzzy logics [2], PID tuning methods [3], sliding mode control [4], and a combination of them [5] are among common control strategies.

The path planning problem is mostly studied in the literature, neglecting the reference velocity planning. This leads to an unsmooth trajectory tracking [6]. However, tracking the desired velocity is of high significance in carrying applications. The carrying mobile robots are widely exploited in health care, agriculture, food preparation, manufacturing, and military industries. An example

of using this technology is the food delivery navigation system [7] and robots carrying the medical rehabilitation devices [8]. It is important to prevent the packages from being exposed to motion forces when carrying food or liquid. To this end, the desired scenario here is to maintain a constant velocity in most of the route. The fuzzy logic method is then exploited to control the mobile robot. It provides the desired control characteristics just by interpreting demanded features to proper logic rules. Few previous works are dedicated to designing the proper fuzzy logics that guarantee velocity tracking. In this regard, the velocity tracking feature has been investigated in specific applications like food delivery robots [7], robots carrying rehabilitation devices [8], and the under actuated finger of a robotic hand [9]. Typically, posture errors are entered into a fuzzy logic for path tracking purposes. However, for a velocity tracking observation, the rate changes of posture errors are also entered to the fuzzy logic [7–9]. The same combination of errors and errors rate are exploited

✉ Elham Almodarresi, e.modarresi@gmail.com | ¹Ariana Technologist Company, Yazd 8917349359, Iran.

as fuzzy inputs in [10], where a fuzzy ring network method is used to control a multi-motor coordinated system and in [11], in which the trajectory tracking of a pneumatic driven linear delta robot is studied. Naturally, the fuzzy system complication expands by increasing the number of inputs. As a solution, a kinematic-based simple technique is innovated in this paper to optimize the number of required fuzzy rules through the Mamdani fuzzy method. In this technique, the velocity tracking is satisfied while no rate change of posture error is entered into the fuzzy logic.

The remainder of the paper is organized as follows. The smooth velocity design is studied in Sect. 2. A kinematic-based fuzzy approach is presented in Sect. 3. The results are presented, compared, and discussed in Sect. 4. The paper is concluded in Sect. 5.

2 Smooth velocity planning

Path smoothing methods like interpolation-based method and path smoothing using special curves (like spline, parabola and ellipse) are introduced in the literature, [12]. Here, the Dubin's curve method, [13] is used for path smoothing. However, instead of the minimal length curve problem considered in Dubin's method, the corner arcs have a constant length, computed using the user-defined maximum angular velocity (ω_m) at corners.

The main goal is to design the reference trajectory such that a constant user-defined velocity is preserved between the rising and falling parts of the velocity graph. Furthermore, the rising and falling parts of the reference velocity profile must have no extremums (Fig. 1).

Denote t_s and v_s as the start time and start velocity and t_f and v_f as the end time and end velocity for the rising or falling parts of the graph. To preserve the continuity of acceleration, $a_s=0$ and $a_f=0$ are assumed. The whole traveled distance by robot is denoted by d_m . A third ordered polynomial describes the acceleration in rising or deceleration in falling part as,

$$A(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3. \quad (1)$$

The following matrix equation is constructed by considering four conditions $A(t_s)=0$, $A(t_f)=0$, $v(t_f)=v_f$ and $d(t_f)=d_m$:

$$B = M^{-1} * [a_s \ a_f \ v_f - v_s \ d_m - v_s(t_f - t_s)], \quad (2)$$

in which, $B = [b_0 \ b_1 \ b_2 \ b_3]^T$ contains the acceleration polynomial coefficients and matrix M is defined as:

Fig. 1 Reference velocity

$$M = \begin{bmatrix} 1 & t_s & t_s^2 & t_s^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ t_f - t_s & \frac{(t_f^2 - t_s^2)}{2} & \frac{(t_f^3 - t_s^3)}{3} & \frac{(t_f^4 - t_s^4)}{4} \\ \frac{(t_f^2 - t_s^2)}{2} - t_s \Delta t & \frac{(t_f^3 - t_s^3)}{6} - \frac{t_s^2 \Delta t}{2} & \frac{(t_f^4 - t_s^4)}{12} - \frac{t_s^3 \Delta t}{3} & \frac{(t_f^5 - t_s^5)}{20} - \frac{t_s^4 \Delta t}{4} \end{bmatrix}$$

in which $\Delta t = t_f - t_s$. The unknown end time t_f is computed for both rising and falling parts, such that no extremums occur along the rising and falling parts. In other words, by computing the rising and falling time intervals such that the corresponding part of the velocity graph has no extremums, a smooth velocity is guaranteed. To this end, the second derivative of the acceleration is computed from (1). Then, b_2 and b_3 are computed from (2) as

$$\begin{aligned} b_2 &= 84 \frac{v_f}{t_f^3} - 180 \frac{d_{\min}}{t_f^4} \\ b_3 &= 120 \frac{d_{\min}}{t_f^5} - 60 \frac{v_f}{t_f^4} \end{aligned} \quad (3)$$

Setting $D^2 A(t_{l_{ts}, t_f}) < 0$ results in computation of rising time interval as

$$t_f \in \left[\frac{15}{8} \frac{d_c}{v_f}, \frac{15}{7} \frac{d_c}{v_f} \right]. \quad (4)$$

Similarly, $t_f = t_s + dt$ describes the falling end time. By setting $D^2 A(t_{l_{ts}, t_f}) > 0$, the falling time interval dt is obtained equal to the same range computed in (4).

3 Fuzzy controller design

Two-wheeled mobile robots are described using the following kinematics:

Table 1 Weights of $\Delta\theta$ membership functions

$\Delta\theta$	BN	N	Z	P	BP
Indicator	-4	-2	0	2	4

Table 2 Weights of ΔD membership functions

ΔD	Z	N	M	F
Indicator	0	2	4	6

Table 3 Weights of ω_R and ω_L membership functions

ω_R, ω_L	BN	N	Z	P	BP	VP	EP
Indicator	-2	-1	0	1	2	3	4

Table 4 Computed weights for inputs $\Delta D, \Delta\theta$

	BN(-2)	N(-1)	Z(0)	P(1)	BP(2)	VP(3)	EP(4)
BN(-2)	-4, 0	-3, -1	-2, -2	-1, -3	0, -4 (Z, BN)	-1, -5	2, -6
N(-1)	-3, 1	-2, 0	-1, -1	0, -2 (Z, N)	1, -3	2, -4 (N, BN)	3, -5
Z(0)	-2, 2	-1, 1	0, 0 (Z, Z)	1, -1	2, -2 (N, N)	3, -3	4, -4 (M, BN)
P(1)	-1, 3	0, 2 (Z, P)	1, 1	2, 0 (N, Z)	3, -1	4, -2 (M, N)	5, -3
BP(2)	0, 4 (Z, BP)	1, 3	2, 2 (N, P)	3, 1	4, 0 (M, Z)	5, -1	6, -2 (F, N)
VP(3)	1, 5	2, 4 (N, BP)	3, 3	4, 2 (M, P)	5, 1	6, 0 (F, Z)	7, -1
EP(4)	2, 6	3, 5	4, 4 (M, BP)	5, 3	6, 2 (F, P)	7, 1	8, 0

$$D(i) = D(i-1) + T \frac{r}{2} (\omega_R(i-1) + \omega_L(i-1)), \quad (5)$$

$$\theta(i) = \theta(i-1) + T \frac{r}{L} (\omega_R(i-1) - \omega_L(i-1)), \quad (6)$$

in which, r and L are, respectively, wheel's radius and the distance between two wheels. Furthermore, $D(i)$ is the total robot's movement up to i th step and $\theta(i)$ is its direction at i th step. The right and left wheels angular velocities are also denoted by ω_R and ω_L .

The inputs are the distance between the robot's present position and the next reference position (ΔD), and the difference between the robot's current orientation and the direction toward the next reference position ($\Delta\theta$). The inputs and outputs are described by triangular membership functions. The distance functions are linguistically named as Zero, Near, Medium and Far. Similarly, five membership functions are linguistically described as Big Negative (BN), Negative (N), Zero (Z), Positive (P) and Big Positive (BP) to describe the orientation error. The linguistic functions attributed to ω_R and ω_L are Big Negative (BN), Negative (N), Zero (0), Positive (P), Big Positive (BP), Very Big Positive (VP) and extremely Big Positive (EP). The Mamdani fuzzy method is employed and the center of gravity method is used

for defuzzification. A kinematic-based approach is presented in proceed for defining the appropriate fuzzy rules. A direct relationship rules between $\omega_R + \omega_L$ and displacement error (ΔD) on one hand and between orientation error ($\Delta\theta$) and $\omega_R - \omega_L$ on the other hand from (5) and (6). Fuzzy rules must be sought in a way that when ΔD and $\Delta\theta$ values alternate between their membership functions, ω_R and ω_L vary in accordance, such that the reference velocity is tracked. To this end,

Table 5 Fuzzy rules for the couple ω_R, ω_L

	Z	N	M	F
BP	BP, BN	VP, N	EP, Z	EP, BP
P	P, N	BP, Z	VP, P	EP, BP
Z	Z, Z	P, P	BP, BP	VP, VP
N	N, P	Z, BP	P, VP	BP, EP
BN	BN, BP	N, VP	Z, EP	BP, EP

some weights are allocated to each linguistic variable (Tables 1, 2 and 3).

Assume ω_R and ω_L are both BP. The weight of BP function is indicated by 2 from Table 3. Now, considering the direct relation between ΔD and $\omega_R + \omega_L$ in (5), ΔD can be notified by the weight 4, which represents "M" from Table 2. Similarly, considering the direct relation between $\Delta\theta$ and $\omega_R - \omega_L$ in (6), $\Delta\theta$ is described by the weight 0 (representing "Z" from Table 1). This builds the corresponding fuzzy rule as: if ΔD is M and $\Delta\theta$ is Z; then ω_R and ω_L are BP. By employing this procedure for all combinations of ω_R and ω_L shown in Table 4, the corresponding weights of ΔD and $\Delta\theta$ are computed simply based on the inverse kinematics. Two subsequent weights, written in each cell, represent ΔD and $\Delta\theta$ in order. The fuzzy rules are summarized in Table 5.

4 Results and discussion

A robot is required to move thorough the defined corners (Fig. 2). The inputs ΔD , $\Delta\theta$ and the outputs, respectively, vary in ranges $[0, 20\text{ cm}]$, $[-\pi, \pi]$ and $[-10, 20]$. Figure 3 shows the velocity tracking by using the proposed fuzzy rules. The initial condition is considered with a 20 cm deviation and $\pi/6$ angular deviation from the reference posture. While a disturbance is encountered at 5th s, the settling time is 2.51 s. Figure 4 compares the results presented here with the results of employing the method in [2]. Undesirable velocity tracking of [2] shows that receiving posture errors as fuzzy inputs do not necessarily lead to velocity tracking while their derivatives are not entered to the fuzzy logic. The velocity graph computed using the method in [2] has 453.16% maximum overshoot from the reference velocity, while this overshoot is 3.69% using the presented method. On the other hand, if the errors derivatives are entered to the fuzzy logic [7–11], velocity is tracked at the price of a larger number of enforced fuzzy rules. Compared to 2×20 rules employed in Table 5, 3×49 rules are designed in [8] and [11].

5 Conclusion

The present paper conducts a study on mobile robots' smooth trajectory design and control, considering the primary role of safe delivery. For this purpose, first, the requirements of having a smooth velocity tracking are investigated. The rising and falling time intervals are

Fig. 2 Reference tracking

Fig. 3 Velocity tracking with initial deviation and disturbance

computed theoretically to prevent extremums in the velocity profile. Then, a kinematic-based algorithm is suggested to extract the proper fuzzy logic. In this method, some weights are assigned to fuzzy logic inputs and outputs. Each weight is already linked to a linguistic function. Then, by substituting the corresponding weights of the outputs into the inverse kinematic equations, the suitable weights are extracted for inputs. In contrast to the literature, although no rate changes of posture errors are filtered through the fuzzy rules, the suggested kinematic-based fuzzy method leads to a perfect velocity tracking.

Fig. 4 Comparing the velocity tracking of the presented method with [2]

Acknowledgments This research was partially supported by Yazd Ariana Technologist Co., Yazd, Iran.

Data availability Available upon the request.

Compliance with ethical standards

Conflict of interest The authors declare that they have no competing interests.

Code availability Available upon the request.

References

1. Koubaa A, Bennaceur H, Chaari I, Trigui S, Ammar A, Sriti MF, Alajlan M, Cheikhrouhou O, Javed Y (2018) In: Introduction to mobile robot path planning. Springer, pp 3–12
2. Mac T, Copot C, Keyser R, Tran T, Vu T (2015) MIMO fuzzy control for autonomous mobile robot. *Int J Control Autom* 3:65–70
3. Kumar UD, Nisha M, Mathivanan N (2018) Tracking of a PID driven differential drive mobile robot. *Int J Mechatron Electr Comput Technol* 8:3690–3704
4. Doria-Cerezo A, Biel D, Olm JM, Repecho V (2019) Sliding mode control of a differential-drive mobile robot following a path. In: 18th ECC, Napoli, Italy, pp 4061–4066
5. Wu H, Karkoub M (2019) Hierarchical fuzzy sliding-mode adaptive control for the trajectory tracking of differential-driven mobile robots. *Int J Fuzzy Syst* 21:33–49
6. Sprunk C, Lau B, Pfaff P, Burgard W (2011) Online generation of kinodynamic trajectories for non-circular omnidirectional robots. In: Proceedings of the IEEE ICRA, Shanghai, China, pp 72–77
7. Sun YG, Guan L, Chang Z, Li C, Gao Y (2019) Design of a low-cost indoor navigation system for food delivery robot based on multi-sensor information fusion. *Sensors* 19(22):4980
8. Zhou Z, Meng W, Ai Q, Liu Q, Wu X (2013) Practical velocity tracking control of a parallel robot based on fuzzy adaptive algorithm. *Adv Mech Eng* 2013:1–11
9. Rakovic M, Anilc G, Mihajlovi Z, Savic S, Naikc S, Borovaca B, Gottscheber A (2018) Fuzzy position-velocity control of underactuated finger of FTN robot hand. *J Intell Fuzzy Syst* 34:2723–2736
10. Wu Y, Cheng Y, Wang Y (2020) Research on a multi-motor coordinated control strategy based on fuzzy ring network control. *IEEE Access* 8:39375–39388
11. Li IH, Chiang HH, Lee LW (2020) Development of a linear delta robot with three horizontal-axial pneumatic actuators for 3-DOF trajectory tracking. *Appl Sci* 10:3526
12. Ravankar A, Ravankar A, Kobayashi Y, Hoshino Y, Peng CC (2018) Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges. *Sensors* 18:3170
13. Dubins LE (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am J Math* 79:497–516

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.